Project: [Wikipedia Website](https://qasvus.wixsite.com/ca-marketing)

TEST PLAN

Project created by: ***Adila Abdullayeva & Fuad Karimov***

Project Manager: ***Sergey Efremov***

1. **Introduction**
2. **Scope**
3. **Main Test Plan**: ***Wikipedia Website testing***
   1. ***Website Manual Testing***
   2. ***Website Automation Testing***
   3. ***Website Automation API Testing***
   4. ***Website Automation Performance Testing***
   5. ***Website Automation Security Testing***
4. **Test Schedule and Estimated Time Approvals**
5. ***Introduction***

We have created the Test Plan to ensure the quality of the Wikipedia (https://www.wikipedia.org/) website. Test Plan helps our team talk and explains how we test everything from created account to logging in to the website. This document includes the strategies and methodologies for functional end-to-end tests. It includes sub-tasks for each test, testing types, environments, and tools and provides test cases with expected and actual results.

1. ***Scope***

The document primarily focuses on the following testing areas:

1. **GUI Testing**: Evaluating the graphical user interface of the application.
2. **Unit Testing**: Testing individual units or components of the software.
3. **Functional Testing**: Assessing the system's functionality, including positive and negative scenarios.
4. **Manual Testing**: Hands-on testing executed by testers.
5. **Automation Testing**: Testing conducted through automated test scripts.
6. **API Testing**: Evaluating the application's APIs.
7. **Performance Automation Testing**: Using automated tools to assess the system's performance.
8. **Security Automation Testing**: Testing the system's security features with automated tools.
9. ***Main Test Plan***: ***Wikipedia Website Testing***
   1. ***Website Manual Testing***

Perform manual testing for the User's Personal Account Address Menu using these testing methods:

1. **GUI Testing:** This means checking how the website looks and feels. We're looking for errors, spelling mistakes, and if it follows design rules.
2. **Exploratory Testing:** Here, we explore the website without predefined test steps. We're trying things out as we go.
3. **Functional Positive Testing:** We want to ensure the website works correctly when we do things correctly.
4. **Functional Negative Testing:** We're also testing what happens when we do things incorrectly or give the website the wrong information.
5. **Functional Ad-hoc Testing:** This informal testing is where we try to find any unexpected problems on the website.

***For our testing, we are using the following environment***:

* **Hardware:** Laptop
* **Operating System:** Windows 11 Home
* **Browsers (latest versions):**
  + Google Chrome
  + Firefox Mozilla
  + Microsoft Edge

We have also listed the results of some test cases in the table below, along with the browsers used and whether the tests passed or failed. These test cases cover both positive and negative scenarios.

***Positive test***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TC ID: | Test Title | Expected Result | Actual  Result | Status Pass/Fail |
| 1 | 2 | 4 | 5 | 6 |
| TC 001 | Verify that the Google search field can open the https://www.wikipedia.org page | Website https://www.wikipedia.org should open | Website https://www.wikipedia.org is opened | PASS |
| TC 002 | Verify that you open the https://www.wikipedia.org page and find the Page LOGO. | User should be able to see the WIKIPEDIA logo | WIKIPEDIA logo is seen | PASS |
| TC 003 | Verify that language buttons are clickable and direct to the correct page | All language buttons should be clickable and direct to their respective page | All language buttons are clickable and direct to correct pages | PASS |
| TC 004 | Verify that the "Search" field is working appropriate | New page with the suggested variants of the typed word is opened | New page with the suggested variants of the typed word is opened | PASS |
| TC 005 | Validate that the proper webpage has opened | User should be on the correct page | User is on the correct page | PASS |
| TC 006 | Verify that the registration is available for user with a valid Email and password | User should be able to open profile after registration | User's profile is opened. | PASS |
| TC 007 | Verify that user successfully logged in with valid Email and password | User should be able to Log-in to his account | User's successfully Log-in | PASS |

***Negative test***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TC ID: | Test Title | Expected Result | Actual  Result | Status Pass/Fail |
| 1 | 2 | 4 | 5 | 6 |
| TC 001N | Verify that the user should be able to get the information about the incorrect login when he/she enters a valid username and invalid password. | The user should get the warning about an invalid username or password | User sees message: "Incorrect username or password entered. Please try again." | PASS |
| TC 002N | Verify that the user should be able to get the information about incorrect login when he/she enters an invalid username and valid password. | The user should get the warning about an invalid username or password | User sees message: "Incorrect username or password entered. Please try again." | PASS |
| TC 003N | Verify that the user should be able to get the information about the incorrect login when he/she enters an invalid username and invalid password. | The user should get the warning about an invalid username or password | User sees message: "Incorrect username or password entered. Please try again." | PASS |
| TC 004N | Verify that the user cannot create an account password that is less than 8 characters | The user should get the warning to choose a password with at least 8 character | User sees message: "Password must be at least 8 characters." | PASS |

***Testing Tools Used:***

|  |  |
| --- | --- |
| **Process** | **Tools** |
| Test case creation | Microsoft Word, Microsoft Excel, JIRA |
| Test case tracking | Microsoft Word, Microsoft Excel, JIRA |
| Test case execution | Manual |
| Test case management | Microsoft Excel, JIRA |
| Test reporting | Microsoft Excel, JIRA |

* 1. ***Website Automation Testing***

Website automation tests have been created using the Selenium Python UnitTest framework based on functional manual test cases.

***Environment Support:***

* **Hardware:** Laptop
* **Operating System:** Windows 11
* **Browsers (latest versions):**
  + Google Chrome
  + Firefox Mozilla
  + Microsoft Edge

***Testing Tools used:***

|  |  |
| --- | --- |
| **Process** | **Tools** |
| Test Case Creation | PyCharm, Selenium |
| Test Case Tracking | PyCharm, Browsers |
| Test Case Execution | Automation |
| Test Case Management | PyCharm |
| Test Reporting | HTML reports, Allure reports |

***We also run additional automation tests using the cloud platform “BrowserStack.”***

***Environment Support:***

* **Hardware:** Laptop
* **Operating System:** Windows 11
* **Browsers (latest versions):**
  + Google Chrome
  + Microsoft Edge

***Testing Tools used:***

|  |  |
| --- | --- |
| **Process** | **Tools** |
| Test Case Creation | PyCharm, Selenium |
| Test Case Tracking | Browserstack.com |
| Test Case Execution | Automation |
| Test Case Management | PyCharm |
| Test Reporting | Video files (mp4) |

The test results align with those obtained on a local device.

* 1. ***Website API Testing***

API tests are conducted to assess the developed APIs' functionality, performance, reliability, and security to ensure they meet website expectations.

***Environment Support:***

* **Hardware:** Laptop
* **Operating System:** Windows 11
* **Browsers (latest versions):**
  + Google Chrome
  + Firefox Mozilla
  + Microsoft Edge

***Testing Tools used:***

|  |  |
| --- | --- |
| **Process** | **Tools** |
| Test Case Creation | Postman, Chrome DevTools-Postman Interceptor |
| Test Case Tracking | Postman |
| Test Case Execution | Automation |
| Test Case Management | Postman |
| Test Reporting | Screenshots |

* 1. ***Website Automation Performance Testing***

Performance automation tests are conducted to measure the tested website's speed, responsiveness, and stability. These tests are performed in incognito environments, and reports are generated after testing.

***Testing Tools used:***

|  |  |
| --- | --- |
| **Process** | **Tools** |
| Test Case Creation | Google Lighthouse, GTMetrix, BrowserStack-SpeedLab |
| Test Case Tracking | Google Lighthouse, GTMetrix, BrowserStack-SpeedLab |
| Test Case Execution | Automation |
| Test Case Management | Jira |
| Test Reporting | Text (.txt), HTML (.html), PDF (.pdf) files |

* 1. ***Website Automation Security Testing***

Security automation tests are conducted to identify potential flaws or weaknesses in software and websites. The testing focuses on assessing whether the application is designed and configured correctly. These tests are also executed in incognito environments, and reports are generated after testing.

***Testing Tools used:***

|  |  |
| --- | --- |
| **Process** | **Tools** |
| Test Case Creation | Mozilla Observatory, PentestTools-Website Scanner |
| Test Case Tracking | Mozilla Observatory, PentestTools-Website Scanner |
| Test Case Execution | Automation |
| Test Case Management |  |
| Test Reporting | Text (.txt), HTML (.html), PDF (.pdf) files |

1. **Test schedule and estimated time**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sub-Task Name** | **Start** | **Est.time**  **(hours)** | **Finish** |
| Manual Testing | 10/28/23 | 1 | 10/28/23 |
| Automation Testing | 11/6/23 | 20 | 11/8/23 |
| API Testing | 11/8/23 | 15 | 11/10/23 |
| Performance Testing | 10/21/23 | 2 | 10/21/23 |
| Security Testing | 10/21/23 | 2 | 10/21/23 |